

IN THE SPECIFICATION:

Please amend the paragraph beginning at page 8, line 5, as follows.

As previously indicated, a component 110 can only invoke services of other components 110 via its own output ports 114. In the present example, ~~it~~ it is important to remember that the not() service is available at the output port 114 only if the output port 114 was previously connected to a component 110 that provides the not service.

Please amend the paragraph beginning at page 12, line 11, as follows.

In the programming environment of the present invention, new instances of a component 110 are created using the "new" command. For example, the statement `MyComponent myComp = new MyComponent()` creates an instance of the component 110 by invoking the constructor `MyComponent()`. The keyword "This" is used as the reference to the current component instance. As used herein, the term "one component interacts with another component," generally means component instances. In other words, a component instance must first be created before methods at the output port can be connected and invoked. But, sometimes we need the ability to invoke certain operations even prior to creating component instances, such as the ability to initialize a component 110. A component 110 can have a special initialization function, `init()`, that that can be directly called by the client component without instantiating a component instance. An `init()` function can take a set of typed parameters and can return a typed value.

Please amend the paragraph beginning at page 20, line 16, as follows.

Generic types allow one to define a "general" type that can be instantiated to more a specific type. Generic types are useful to define collection components. Generic components ~~consists~~ consist of a port whose type is a generic type. Genericity is based on F-bounded parametric genericity. A generic stack component is constructed as follows. First, `StackInterface` is defined as follows:

Please amend the paragraph beginning at page 23, line 21, as follows.

The input port bci of the Container component is a contract. The port bci is attached to the component BooleanComp that implements the contract type BooleanCont. The client component first creates an instance of the Container component, and connects to it via bcout port. It then connects the bio to bcout.bin. Finally, it invokes the bio.not() service. It is noted that the client component is once again unaware of the component within Container that implements the contract BooleanCont.

Please amend the paragraph beginning at page 25, line 23, as follows.

FIG. 4 outlines the identifiers used by the present invention. It is assumed that names for variables, components, contracts, interfaces, classes, fields and methods are distinct.